

Scalable Architecture of MIMO Multi-carrier CDMA System on Programmable Logic

Yuanbin Guo

Research, Technology and Platform,
Nokia Siemens Networks,
Irving, TX, 75039
Email: ybguorice@gmail.com

Joseph R. Cavallaro

Electrical and Computer Engineering,
Rice University,
Houston, TX, 77005
Email: cavallar@rice.edu

Abstract—In this paper, a scalable architecture of the multi-carrier CDMA system using Multiple-Input-Multiple-Output (MIMO) technology is designed in the programmable logic array. The system-level partitioning with different architecture design entries is described. The overall computing architecture for complex signal processing blocks, e.g., channel estimation, frequency domain equalization, demodulation etc is described. The MIMO architecture is easily extended from a SISO system with single antenna. This scalable architecture demonstrates resource utilization efficiency and easy extension to MIMO configurations.

I. INTRODUCTION

Much higher data rate than what is offered today is required for multimedia services and ubiquitous networking via mobile devices with the emerging beyond 3G and 4G wireless communication technologies. Due to the excellent performance in dealing with hostile frequency selective wireless channels, Orthogonal Frequency Division Multiplexing (OFDM) technologies have been researched extensively for many different standards. Multi-carrier CDMA [1] is one key technology that combines OFDM and CDMA because of its many advantages, such as larger capacity, high data rate support etc. On the other hand, the use of MIMO (Multiple Input Multiple Output) technology [2] is becoming more and more important because of its capability to enhance the spectrum efficiency significantly. Recent years have seen the combination of MIMO with the Multi-carrier CDMA as an important candidate for the emerging B3G and future 4G wireless systems [4].

However, these technologies also involve many complex signal processing algorithms, which demand tremendous processing power to achieve the real-time performance with low-cost silicon usage and low power consumption. Due to the advances in silicon technologies, many very complex signal processing algorithms are now feasible to be implemented in dedicated DSP processors. However, the increase rate of processing power for most commercial DSP processors is still behind the complexity increase in the demanding signal processing algorithms, especially for the MIMO MC-CDMA systems. Even with the emerging multi-core System-on-Chip (SoC) DSP processors, many baseband signal processing are still more suitable for hardware acceleration with VLSI circuits.

Field-Programmable Logic Arrays (FPGA) have been typically used for early phase prototyping of these hardware accelerations because of their much higher parallelism compared with processor-type architectures. However, they also have much less flexibility compared with DSP processors. Designing efficient architectures that are scalable for different configurations is usually challenging, especially when the scalability means to support multiple antennas [3]. To reduce redundancy, commonalities in different configurations must be exploited.

In this paper, scalable architecture for the complete PHY layer of a MIMO MC-CDMA system is proposed. We first start with the baseline implementation of the SISO system. Efficient architectures for the dominant signal processing algorithms are designed. The commonality in a MIMO system is studied. Partitioning strategy for both software/hardware and design entries with different flow is described. The tradeoff in data throughput and the design area is exploited to derive the scalability from a SISO system to MIMO systems. We also utilize different design flows for design entries in different domains based on the features of the algorithms. A Catapult C flow [5] is applied to synthesize the design architecture to RTL design. The architecture is implemented in a small form factor demonstrator based on Nallatech design boards with multiple Xilinx Virtex-II FPGAs. The architecture not only demonstrates efficiency in resource utilization, but also has the scalability for easy extension to MIMO configurations.

II. MIMO MC-CDMA SYSTEM MODEL

A simple model of the basic SISO MC-CDMA system is described here, as depicted in Fig. 1. It is basically a serial concatenation of classical CDMA and OFDM systems. First the multi-users' modulator outputs are split into K blocks of J streams. For each block, J Walsh-Hadamard codes of length K are used to spread these J symbols and the output sequences are summed up to form a single chip rate stream. After the spread streams are passed through a serial-to-parallel converter, they are interleaved with chip streams from other blocks. Then some equally spaced pilot tones are inserted into the data sequence by frequency-multiplexing. The parallelized data stream in the frequency domain will then pass through an N_F -point IFFT block to generate the OFDM symbol. After

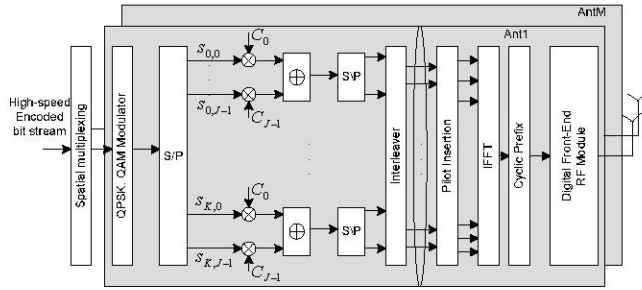


Fig. 1. System Block Diagram of the Multi-carrier CDMA System.

that, the cyclic prefix is added and the resulting time-domain sequence is passed to the Digital-Front-End (DFE) for filtering and other DFE related processing.

By assuming a multi-path channel of L paths, the received signal is a convolution of the transmitted signal with the channel taps with white Gaussian noise added. After passing through the corresponding front-end receive filters, the time domain OFDM symbols are collected and the cyclic prefix is removed from each OFDM symbol. In the frequency domain processing, complex signal processing algorithms are used to detect the received symbols.

For the MIMO transmission, the VBLAST type of spatial multiplexing based open-loop transmission is considered for implementation. The transmitter extension is relatively simple, as we can view it as a multiple parallel processing blocks duplicated from the SISO transmission. Functionally, we can view some jointly encoded high-speed bit streams first spatially de-multiplexed into multiple lower speed data streams. Each stream will have the same processing of the SISO data path. As seen from a simplified block diagram in Fig. 1, the scalability from SISO to MIMO can be viewed as simple duplication of processing elements. However, to derive a more efficient architecture, the processing power in each major blocks need to be balanced and re-partitioned, as we will show later.

The extension from SISO to MIMO at the receiver is much more challenging, as more advanced detection algorithms are needed to do the joint detection for multiple antennas. As this paper will focus on the scalable architecture for implementation, we will not elaborate the mathematical equations too much. But we will focus on the architecture design of the dominant signal processing blocks by addressing the design entry partitioning, commonality analysis and reusability etc. Major equations will be explained as needed to facilitate the understanding.

III. SCALABLE ARCHITECTURE FROM SISO TO MIMO

A. Architecture for SISO Transceiver

The practical implementation of the prototyping system is much more complex than the data flow block diagram shown in Fig. 1. The partitioning of the SISO transmitter on the FPGA is shown in Fig. 2. The partitioning is based on the signal processing requirement, data flow, and the computing

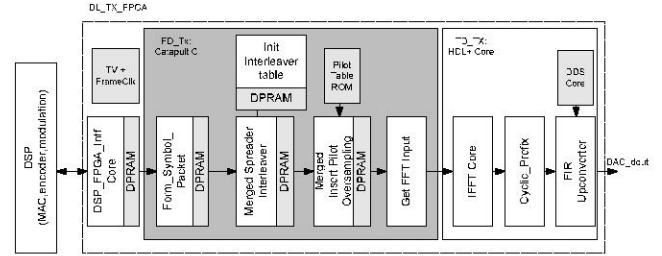


Fig. 2. FPGA Architecture for the SISO Multi-carrier CDMA Transmitter.

architecture. The partitioning between DSP and the FPGA is at the boundary of encoder and the baseband transmitter. First, the DSP will transfer a frame of data to the FPGA buffer. The first block in the FPGA will form an OFDM symbol packet based on the MC-CDMA frame structure. For example, for the first OFDM symbol of each slot, the preamble is inserted at the head. The spreader and the inter-leaver are merged in a single module because the loop structure for each OFDM symbol is similar for the spreader and inter-leaver. To support configurable interleaver, an interleaver table initialization is designed to generate interleaving indices to a memory block. Also the pilot locations are stored in a ROM block to make it configurable. The pilot insertion and some over-sampling functions are merged to prepare the OFDM symbol data for the IFFT. These functions are very suitable for high-level synthesis based design flow. They are designed by the Mentor Graphics Catapult C design tools [5].

The time domain transmitter functions include the IFFT and the FIR filtering as well as digital up-conversion. These functions require more complex signal processing power, but are more or less common standard modules. It is more efficient to integrate off-the-shelf high-performance IP cores. Thus IP cores from 3rd party providers are integrated in the HDL designer environment.

The architecture partitioning of the SISO receiver is shown in Fig. 6. It is also partitioned into three design entry domains. First at the interface of ADC input, some digital down-converter and Low Pass (LP) filter modules are integrated with other modules such as synchronization and frequency offset correction. The low pass filter is used to suppress the out-of-band noise and interference. Together with the cyclic prefix removal and FFT module, they form the processing sub-system in the time-domain. Because many of these blocks require high-throughput processing power, they are more suitable for optimized IP cores designed with HDL entries. In our design, we have integrated both Xilinx CoreGen cores such as FIR filter, digital synthesizer etc. The FFT core is an 1944-point mixed-radix FFT designed by RF engineering. This time-domain front-end requires a lot of sample level operation in streaming data mode.

After the FFT module, the processing is based on the OFDM symbol and thus it is more suitable for block mode operation. These modules form the frequency-domain receiver. A down-sampling module will first follow the FFT. Then a Rx filter compensator compensates the filtering characteristics

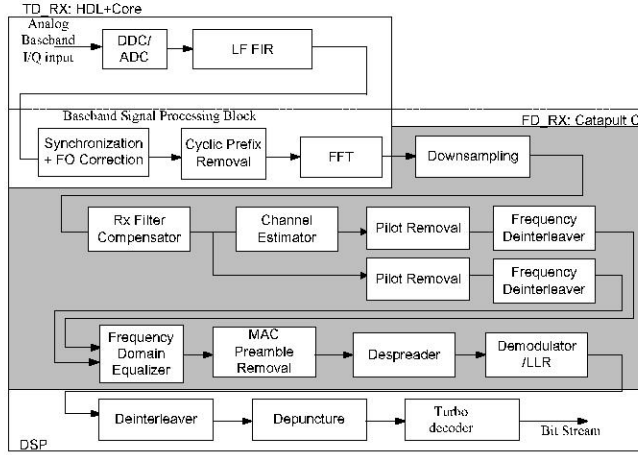


Fig. 3. Partitioning of the SISO Multi-carrier CDMA receiver.

from the pulse shaping filter. A frequency-domain channel estimator block computes the channel coefficients for each tone. After removing the pilots and after the frequency-domain deinterleaver, the signal is passed to a frequency-domain equalizer to recover the signal from the channel distortion. After removing the MAC and preamble information, the data portion package is de-spread and passed to the QAM demodulator. Here the LLR information is calculated and passed to the DSP for channel decoding. After the synchronization point for each symbol is acquired, the frequency-domain operation is deterministic. However, because of the many proprietary receiver algorithms, we use Catapult C design all the functions in the frequency domain modules. They are integrated with the time-domain RX subsystem in the HDL environment.

B. MIMO Transceiver Architecture

1) *MIMO Transmitter Scalable Architecture*: With the description of the SISO TX FPGA architecture, we can scale the design to support the MIMO configuration based on the different throughput requirement and design area for each module. First option is shown in Fig. 4. In this option, the frequency domain FD_TX module is scheduled by Catapult C to be a very area-efficient design, which just supports the throughput requirement of a SISO system. Because of the small design area of this design, it is very easy to duplicate the design entities for a MIMO system. However, for the IFFT module, since it is designed with very high throughput by the third party, it is a relatively large design with tremendous pipelining and parallelism already. The throughput of the IFFT alone is sufficient for the targeted MIMO configuration. Thus, we can first split the data with spatial multiplexing to multiple parallel SISO FD_TX , and the time multiplex the IFFT module. The SISO TD_TX frontend is already designed to be as small as possible for a single antenna. Also because of the sample level processing in front-end, it is more suitable to separate the processing for different antennas. Thus, we split the output from the IFFT for different stream into separate TD_TX frontend.

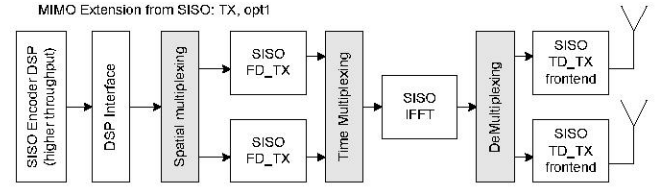


Fig. 4. FPGA Architecture of the MIMO Multi-carrier CDMA Transmitter: Option 1.

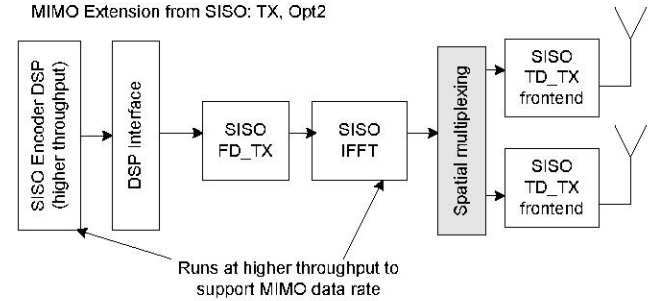


Fig. 5. FPGA Architecture of the MIMO Multi-carrier CDMA Transmitter: Option 2.

Another option is to exploit more pipeline in the SISO FD_TX module. Thus, the throughput for a single SISO FD_TX entity is already sufficient for multiple stream support. As the third party IFFT module is already a highly pipelined design, we simply move the spatial multiplexing to the time-domain boundary, i.e., the TD_TX frontend module. This is shown in Fig. 2. Option 2 has the benefit of less overhead in designing the multiplexing, de-multiplexing interfaces, while put the flexibility into the design of the SISO FD_TX module. By exploiting the block-level pipelining as shown in Fig. 4 with Catapult C, the higher throughput for MIMO support is easily achieved in our design as shown in Fig. 5.

2) *MIMO Receiver Scalable Architecture*: From the functional level partitioning of the receiver in Fig. 3, the detailed FPGA design architecture of the receiver blocks is shown in Fig. 6. First, the pilot index table is stored in ROM block as in the transmitter. Because the pulse shaping filter is also predefined, the frequency domain coefficients are stored in the ROM block to compensate the pulse shaping filter at the transmitter. Moreover, as the pilot extraction has the same loop structure as for the filter compensation module, they are merged as a single module to reduce loop redundancy.

A Least Squares (LS) estimation algorithm is used for estimating the channel. We assume that the impulse response of the channel is \mathbf{h}_c with L_c delay spread. If the cyclic prefix is long enough to cover the delay spread, it is shown that the LS estimation of the channel is given by

$$\hat{\mathbf{h}}_c = \mathbf{V}\mathbf{S}^{-1}\mathbf{U}^H\mathbf{Y}_p. \quad (1)$$

where $\mathbf{U}, \mathbf{S}, \mathbf{V}$ are generated from the SVD of $\mathbf{H}_{tp}\mathbf{W}_p = \mathbf{U}\mathbf{S}\mathbf{V}^H$. Here \mathbf{W}_p is the resulting FFT matrix only dedicated to the pilot symbols, \mathbf{H}_{tp} is some constructed matrix according to the pilot information. Thus, it is given that the frequency

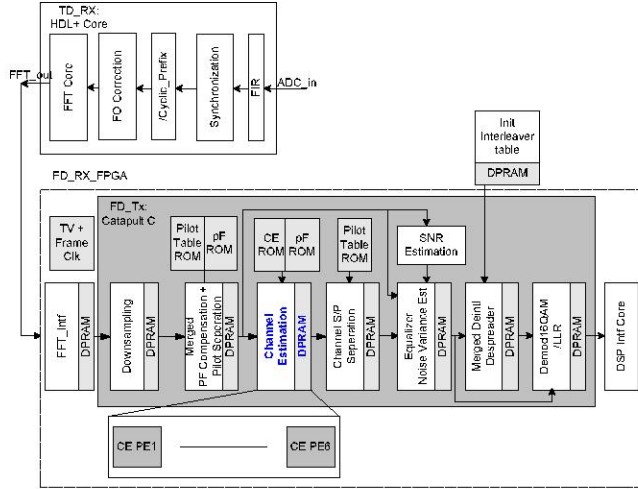


Fig. 6. FPGA Architecture for the SISO Multi-carrier CDMA receiver.

response of the estimated channel $\hat{\mathbf{h}}_c$ is obtained by taking FFT of $\hat{\mathbf{h}}_c$ as

$$\hat{\mathbf{H}}_c = \text{diag}(\mathbf{W}\hat{\mathbf{h}}_c) = \text{diag}(\mathbf{W}\mathbf{V}\mathbf{S}^{-1}\mathbf{U}^H\mathbf{Y}_p), \quad (2)$$

where \mathbf{W} is the DFT matrix. Because all the matrices except the data vector \mathbf{Y}_p to construct the $\hat{\mathbf{H}}_c$ are known a priori, the matrix $\mathbf{L} = \mathbf{W}\mathbf{V}\mathbf{S}^{-1}\mathbf{U}^H$ is computed off-line and stored in the ROM blocks. The channel estimation in the frequency domain is then essentially a matrix-vector multiplication of the predefined matrix in ROM blocks and the frequency-domain input data.

Note that here we need to estimate the channel coefficients for each subcarrier and these sub-carriers are considered independent to each other. We can exploit the parallelism in the subcarrier level to speed up the computation. If we store all the predefined coefficients in a single ROM block, the memory race would cause some data dependency, which stalls the intrinsic parallelism in the algorithm. Thus, we need to split these into sub-block memories and create multiple processing elements for channel estimation. This is shown as the multiple parallel processing elements for channel estimation in Fig. 6.

When the frequency-domain channel coefficients are estimated, the channel is equalized by using an LMMSE algorithm as

$$\hat{\mathbf{X}}_{LMMSE} = \mathbf{H}^H(\mathbf{H}\mathbf{H}^H + \sigma^2\mathbf{I}/P)^{-1}\mathbf{Y}. \quad (3)$$

Where \mathbf{H} is the frequency domain channel matrix for each subcarrier, and \mathbf{X} and \mathbf{Y} are the detected symbol vector and the received signal vector, respectively. σ is the noise variance and P is the transmit power. Thus, we need another block to estimate the SNR as shown in Fig. 6. Note that the equalizer itself is done for each subcarrier independently. Thus, the equalizer is essentially a computation loop for all sub-carriers, where the loop entity does the equalization computation. For the SISO case, the channel coefficient is a scalar for each subcarrier. Similar to the transmitter, it is straightforward to merge

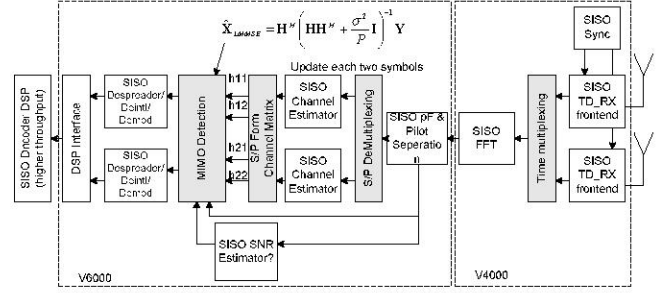


Fig. 7. FPGA Architecture of the MIMO Multi-carrier CDMA Receiver.

the de-interleaver and the de-spreader into a single processing module because of their similarity in the structure.

To identify the scalability in the receiver side, we need to analyze the difference of the SISO vs. MIMO receiver. As it shows, when the LMMSE equalizer is applied, the MIMO architecture has quite many similarity with the SISO receiver. The extension from the SISO receiver to the MIMO receiver architecture is shown in Fig. 7.

The major difference here is that the channel becomes a MIMO channel matrix compared with the SISO case. However, the frontend processing for each antenna is independent. Because of the high sampling frequency rate for the front-end filtering, we can simply duplicate the front-end path for each antenna. The FFT is multiplexed in the same way as the IFFT in the transmitter. Also because the pilot separation and pulse-shaping filter module is easy to achieve high throughput design with relatively small resource utilization, we can also multiplex a single SISO processing element as in the FFT. However, we need to estimate more channel coefficients for each subcarrier as compared with the SISO case, the channel coefficients form a matrix for each independent subcarrier. Even though, these coefficients are decoupled from each other in the estimation. Thus, we can duplicate the SISO channel estimator processing components to meet the increased throughput requirement. Thus, we have both the subcarrier-level parallelism and the antenna-level parallelism.

The actual LMMSE MIMO detector now becomes a joint detection for multiple streams as in the LMMSE detection equation. For each subcarrier, this requires the matrix multiplications and the matrix inversion compared with the scalar multiplication and division in the case of SISO case. The complexity increase thus is not linear to the number of streams. However, for the despreader + deinterleaver/demodulator functions, the SISO design modules can be reused and scaled to support the MIMO configurations.

IV. DESIGN SUMMARY

A. FPGA Resource Utilization

This section summarizes the FPGA resource utilization for the major building blocks. Table I shows the summary of the major building blocks for the MC-CDMA transceiver system. The 1944-point FFT is a mixed-radix pipelined architecture based on multiple processing stages. Each stage has switched

TABLE I
SUMMARY OF THE MAJOR BUILDING BLOCKS FOR THE MC-CDMA
SYSTEM

Module	MULT18x18	Ram16s	Slices
1944.FFT	38	58	10289
DL FD Tx /wo FFT	2	3	3560
DL TD Rx /wo FFT	3	6	2150
DL FD RX	74(61%)	101(80%)	15556(67%)
Total in 2V4000	120	120	23040

TABLE II
BREAK OUT OF THE FPGA COMPONENTS IN THE FD RX

Module	MULT18x18	Ram16s	Slices
pF compensation	4	6	2298
Chan Estimation	48	-	2403
SNR Estimation	13	-	2324
Equalizer	8	-	3626
Demod LLR	1	-	2656

delay elements and some DFT/IDFT computation. A single FFT module consumes 38 multipliers and 58 RAM16 blocks. It also consumes the most number of slices for a single module. However, as it is designed as a continuous pipeline, after the initial delay due to the pipeline filling up. It turns out that a single FFT module can be multiplexed to support 2 antenna processing.

Both the frequency domain and time domain processing modules are summarized in the table as well. As it can be seen, these design modules are much smaller than the FFT module. Thus it is not very costly to duplicate multiple processing elements for the MIMO system.

The front-end of the time domain receiver is similar to the time domain transmitter processing. However, the complete frequency domain receiver design alone requires 74 multipliers and 101 RAM16 blocks. The number of slices is 67% of the Virtex-II V4000 device. The breakup of the different modules in the frequency-domain equalizer is shown in Table. II. The channel estimation module alone consumes the most number of multipliers. After the channel coefficients are obtained for each sub-carrier, the complexity of the equalizer itself is relatively simple. Because of independence between the channel coefficients, the scalability to MIMO channel estimation can be achieved easily by increase the number of processing elements to meet the throughput requirement. As the joint detection for the MIMO detection only changes the internal entity in the loop structure compared with the SISO system, extension to the MIMO system can still maintain the design size of the equalizer well-balanced.

V. CONCLUSION

In this paper, we present a scalable architecture for the MIMO multi-carrier CDMA system. The commonality between the SISO and MIMO systems is exploited for reusability of the major design modules. The design is prototyped in the FPGA platform, which demonstrates architecture efficiency and scalability.

REFERENCES

- [1] S. L. Nours, F. Nouvel, J. F. Helard, *Design and implementation of MC-CDMA systems for future wireless networks*, EURASIP Journal on Applied Signal Processing, pp. 1604-1615, 2004.
- [2] G. D. Golden, J. G. Foschini, R. A. Valenzuela and P. W. Wolniansky, *Detection algorithm and initial laboratory results using V-BLAST space-time communication architecture*, Electron. Lett., Vol. 35, pp.14-15, Jan. 1999.
- [3] Y. Guo, J. Zhang, D. McCain and J. R. Cavallaro, *Scalable FPGA architectures for LMMSE-based SIMO chip equalizer in HSDPA downlink*, 37th IEEE Asilomar Conference, Monterey, CA, 2003.
- [4] Y. Li, J. H. Winters and N. R. Sollenberger, *MIMO-OFDM for wireless communications: signal detection with enhanced channel estimation*, IEEE Transactions on Communications, vol. 50, pp. 1471-1477, Sept. 2002.
- [5] Y. Guo, G. Xu, D. McCain, J. R. Cavallaro, *Rapid scheduling of efficient VLSI architectures for next-generation HSDPA wire-less system using Precision-C synthesizer*, Proc. IEEE Intl. Workshop on Rapid System Prototyping'03, San Diego, CA, pp. 179-185, June 2003.